Jet Propulsion Laboratory
California Institute of Technology

The OpenSE Cookbook: A practical, recipe based collection of patterns, procedures, and best practices for executable systems engineering for the Thirty Meter Telescope

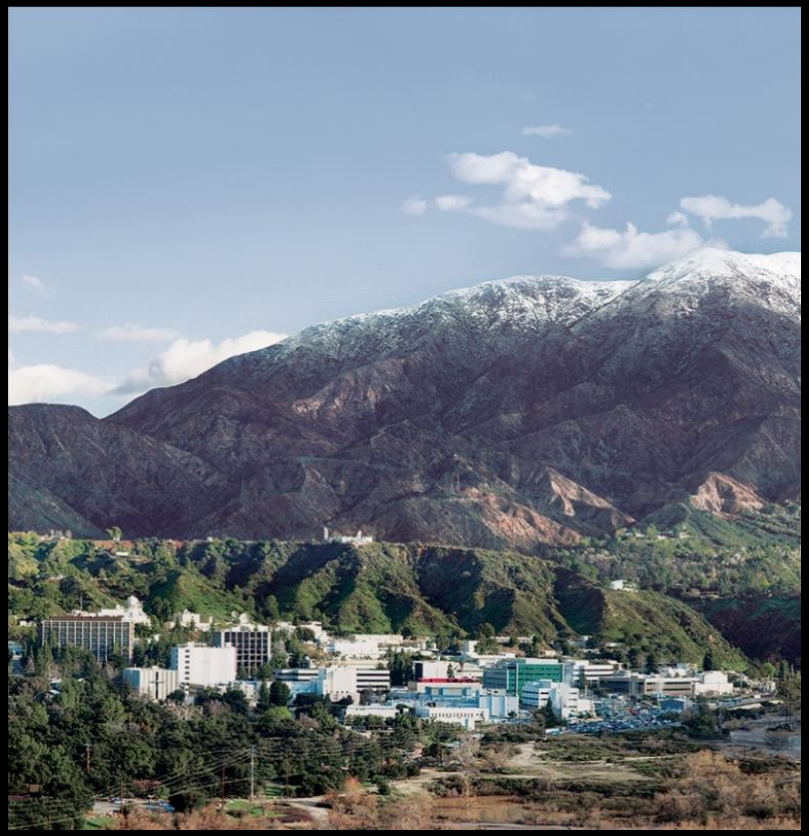
June 12, 2018, Robert Karban, CAE Project Systems Engineer
Jet Propulsion Laboratory,
California Institute of Technology

11-15 June 2018 – SPIE Astronomical Telescopes + Instrumentation, Austin, TX, USA

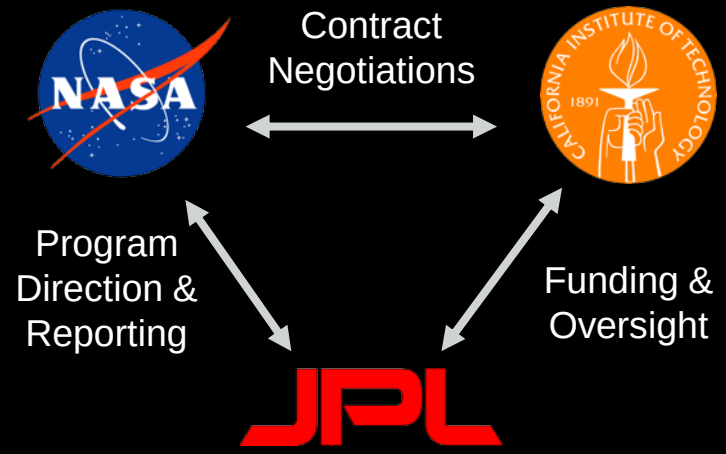
Agenda

- Introduction
- TMT and MBSE Approach
- Cookbook Principles
- Cookbook Examples

NASA Jet Propulsion Laboratory (JPL)

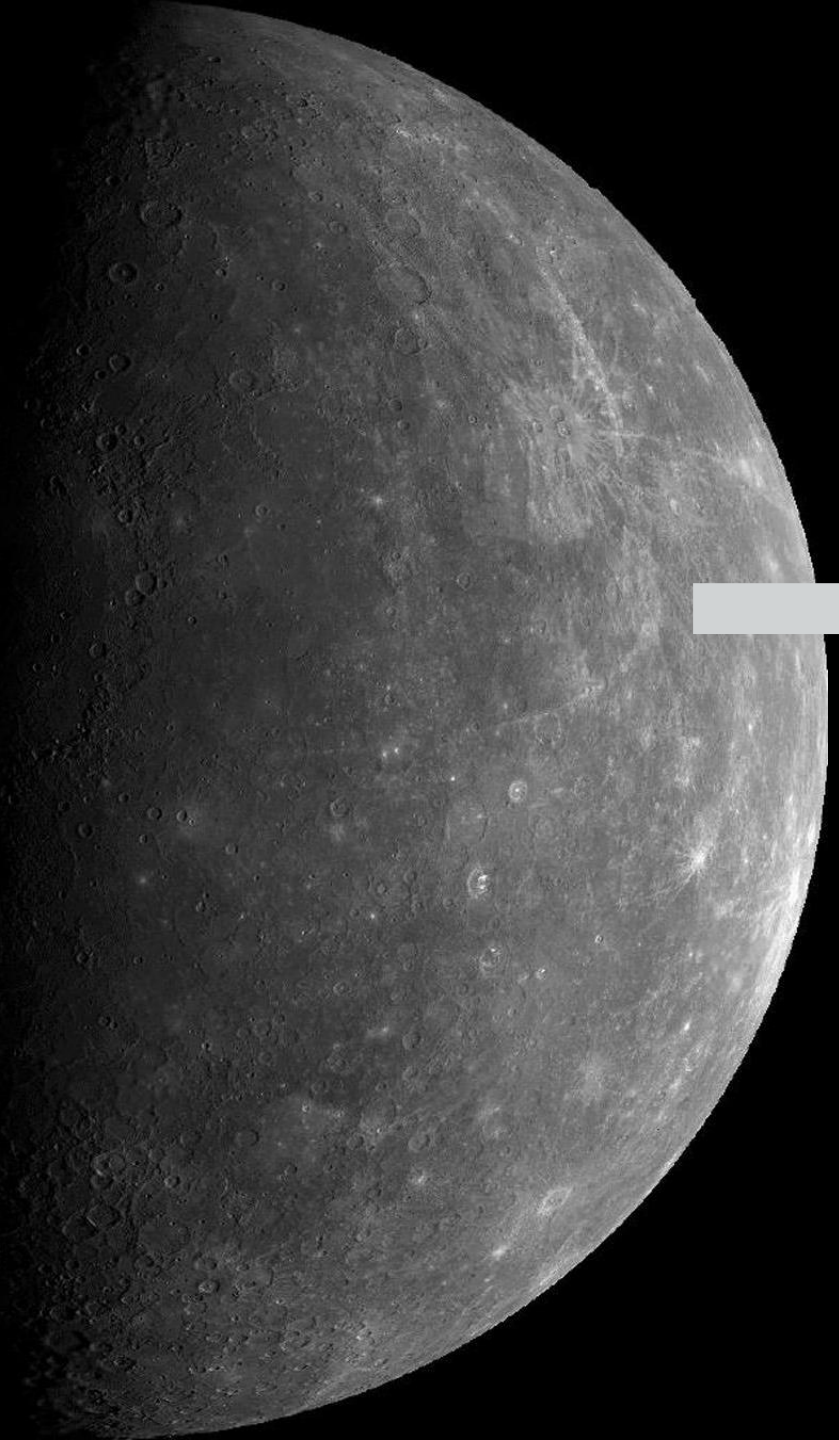


- Located in Pasadena, CA
- NASA-owned *"Federally-Funded Research and Development Center"*
- University-operated
- ~5,000 employees



Who is Robert?

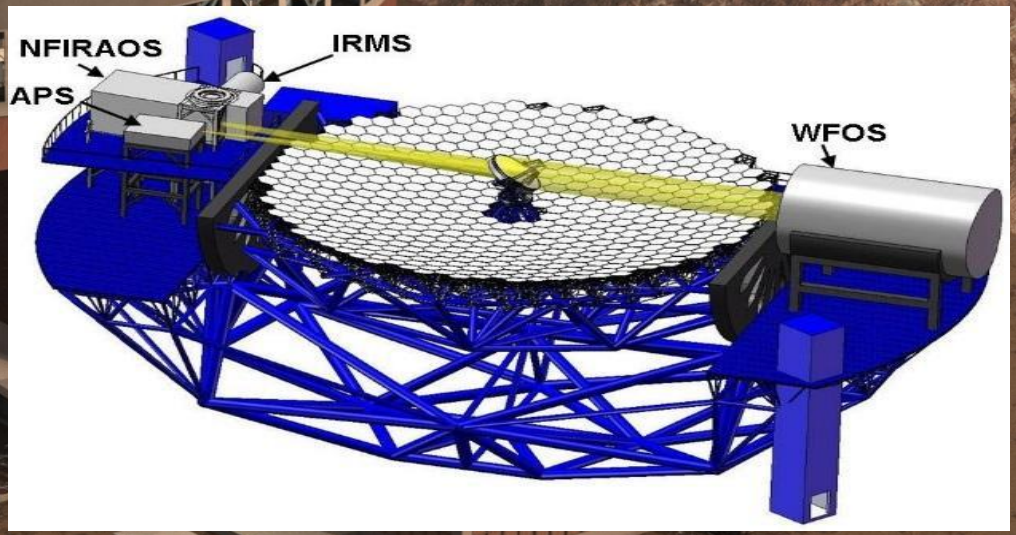
- CAE Project Systems Engineer
  at NASA's JPL - USA
- Member of INCOSE
- Co-Chair of the OMG SysML Revision Task Force
- Formerly Control System/Software Engineer and Architect at:
  - European Southern Observatory – Germany, Chile
  - CERN – Switzerland/France
  - Siemens Healthcare - Austria
- M.Sc. Computer Science (Austria)



Agenda

- Introduction
- TMT and MBSE Approach
- Cookbook Principles
- Cookbook Examples

The Thirty Meter Telescope (TMT) Project



www.tmt.org

TMT applies “Hybrid” Systems Engineering Approach

Traditional SE

- Clear, defined deliverables
- Easily accessible
- Shallow learning curve
- Simple traceability

MBSE

- Understanding behaviors of a system
- “Rich” capability to represent complex systems

Exploit the advantages of each approach

TMT MBSE Approach delivers consistent, verifiable engineering products

- Define an **executable SysML model**
- Use the model to **analyze the system design and verify requirements** on power consumption, mass, duration, pointing errors, etc.
- Produce **engineering documents**
  - Requirement Flow Down Document
  - Operational Scenario Document
  - Design Description Document
  - Interface Control Documents
- Use **standard languages and techniques, and COTS tools where practical** to avoid custom software development

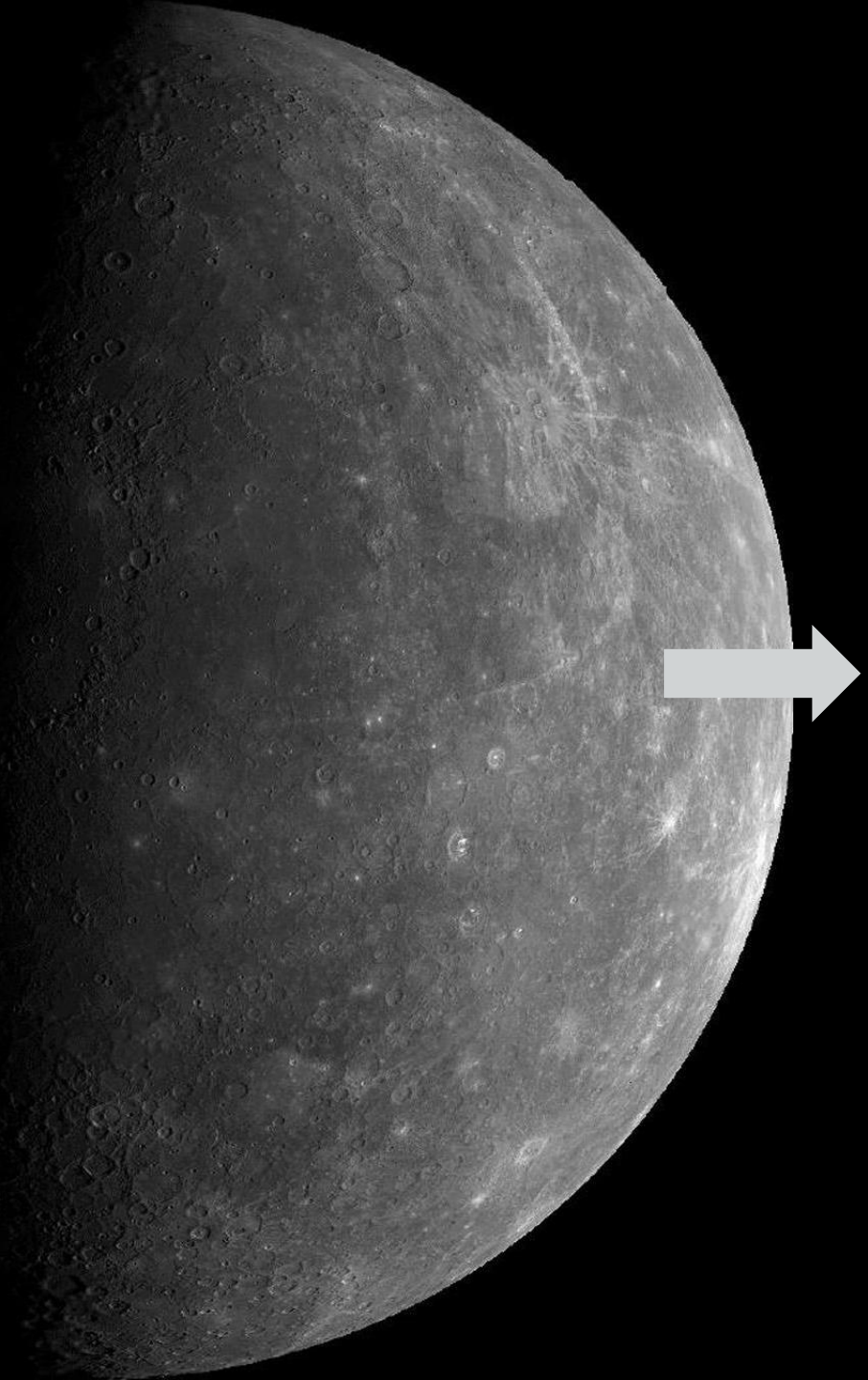
TMT MBSE follows a well defined Modeling Approach

- Object-Oriented Systems Engineering Methodology (**OOSEM**), but with additional activities focusing on building an executable model
- Use case driven model development
- Challenges:
  - JPL is a **supplier** for a number of subsystems of the TMT (the **customer**)
  - Model is used by a number of teams, including TMT

ESEM = OOSEM + Executability

Executable SE Approach focuses on Key SE Artifacts

- Emphasize executable models to enhance understanding, precision, and verification of requirements
- Executable Systems Engineering Method (**ESEM**) augments the OOSEM activities by enabling executable models
  - ESEM defines executable SysML models that verify requirements
  - Includes a set of analysis patterns that are specified with various SysML structural, behavioral and parametric diagrams
  - Also enables integration of supplier/customer models and analysis



Agenda

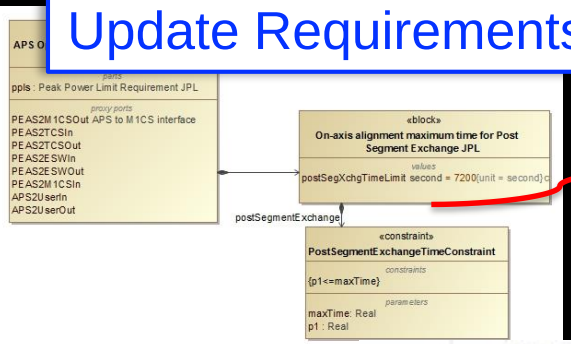
- Introduction
- TMT and MBSE Approach
- Cookbook Principles
- Cookbook Examples

OpenSE Cookbook addresses Systems Engineering Concerns

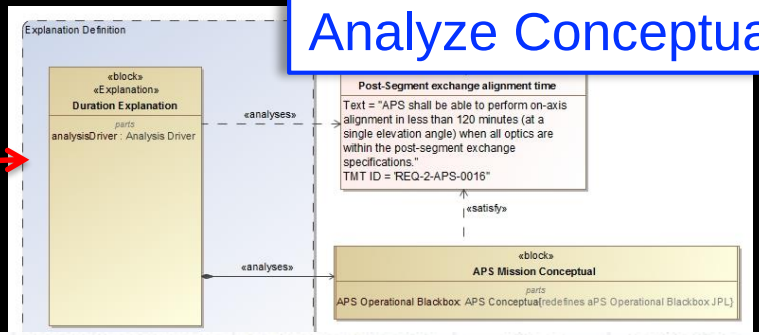
- Provides goal oriented guidance using patterns, e.g.
  - How-to Verify Requirements
  - How-to Roll-up Technical Resources
- Driven by Systems Engineering Workflows
- Enables combing patterns into more complex recipes
- Demonstrates how to build system models with available tooling - How/where do I start?
- Includes known usages in TMT production model as reference
- Commoditizes Executable Systems Engineering

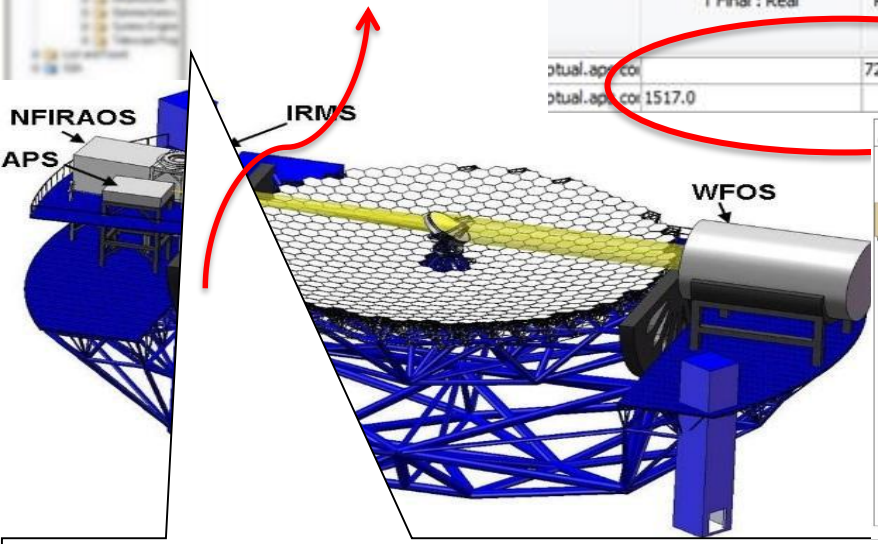
TMT Analysis workflows drive the OpenSE Cookbook

Update Requirements 2

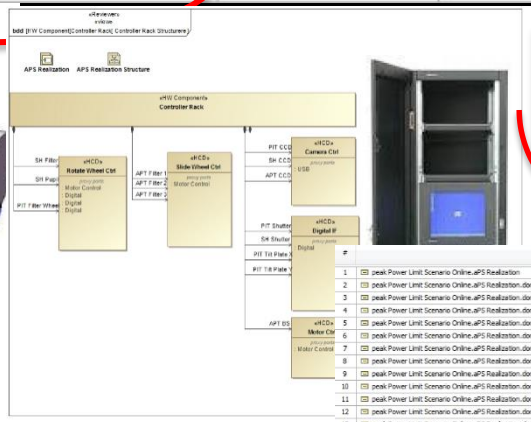


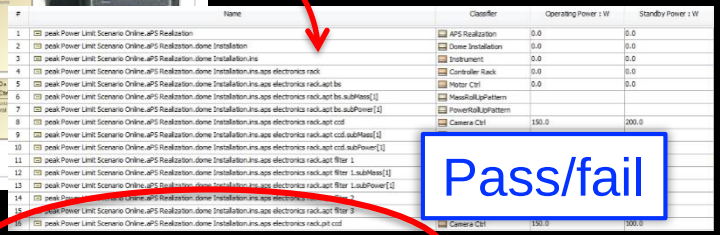
Analyze Conceptual Design 3





Analyze Realization Design/Specification 4





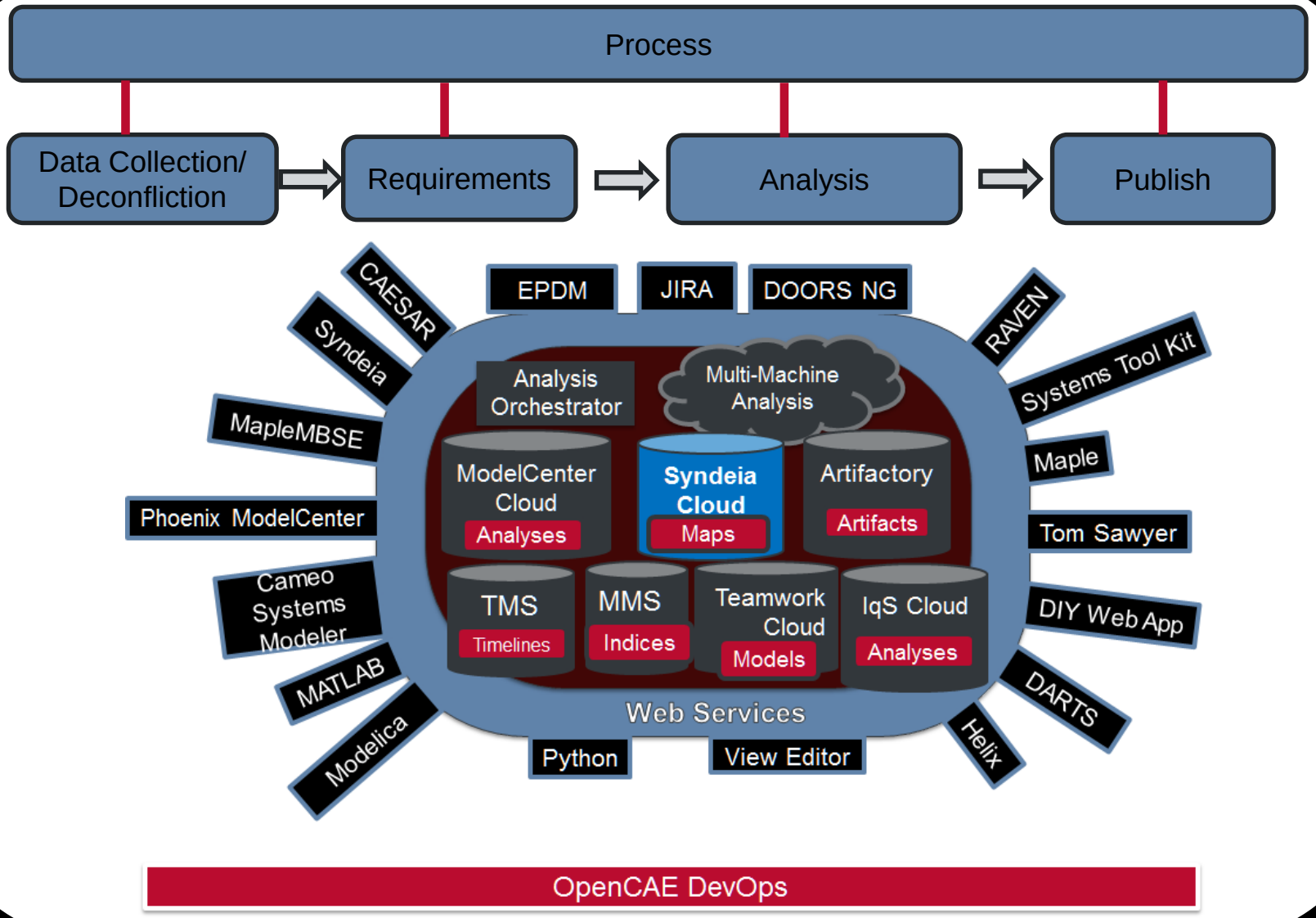
Pass/fail

Produce Engineering Documents 5

Change request 1

Max duration Post-segment exchange: ~~7200s~~ 5000s
Number of exposures of 45s 4 ~~6~~
Max peak power consumption in dome: 8.5kw ~~8.1kw~~
Number of motors with 50W 10 ~~12~~

JPL/CAE Systems Environment provides integrated Life-Cycle Support



JPL develops requirements for Systems Environment (tooling) through Case Studies

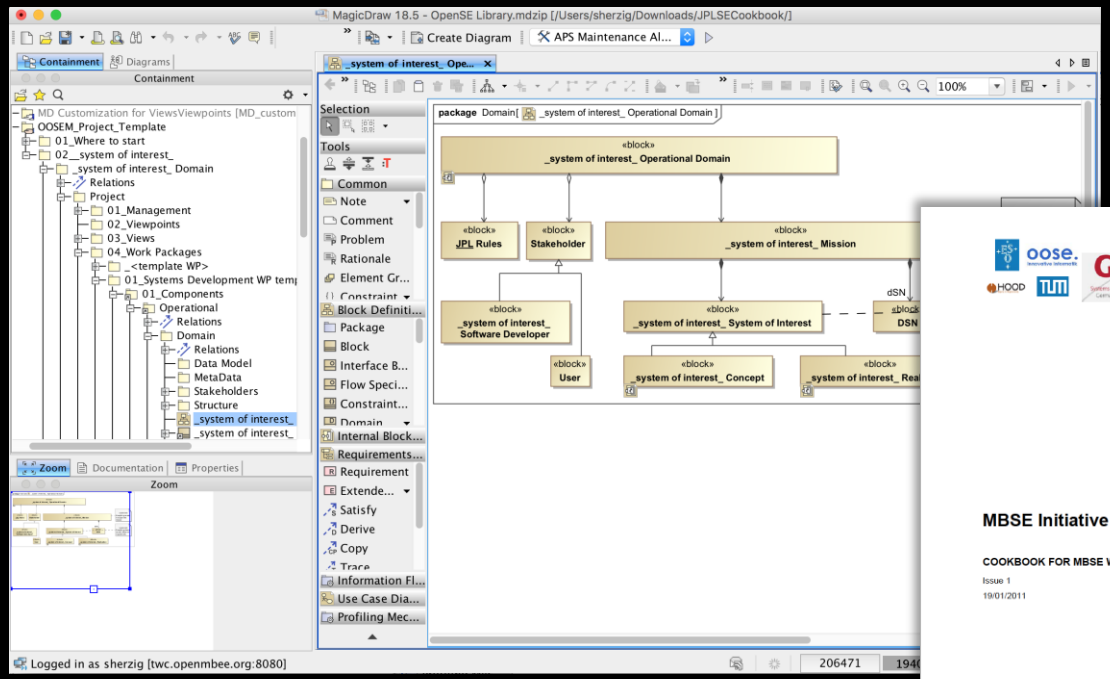
For Example:

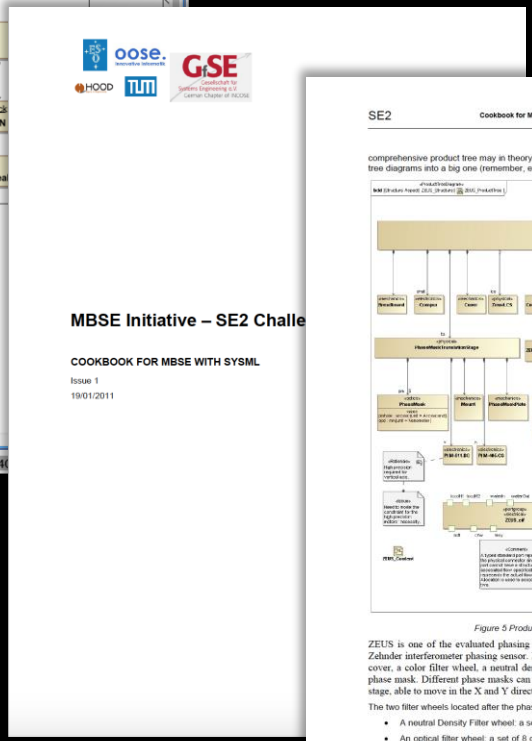
- Requirements Management
- Interface Management
- Design Management
- Trade Studies
- Interdisciplinary Integration
- Analysis Pipeline
- Resource Management
- Timeline Management

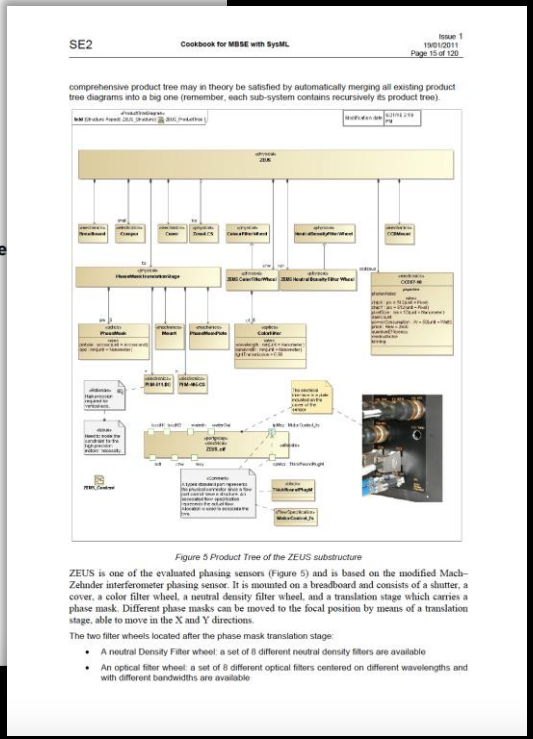
OpenSE Cookbook combines different aspects

- Update 2012 “Cookbook for MBSE with SysML”
  - Focus on structure and requirements using European FP7 Active Phasing Experiment (APE) as case study
- Include Patterns developed for TMT
  - Focus on behavior and analysis workflows
- Guided by ESEM methodology
- Describe tooling support provided by JPL Systems Environment
- OpenSE model library provides commonly used elements
- Instructional examples
- Application to actual engineering team, i.e. TMT
- Template Models and recommended model organizations

OpenSE Cookbook and Template Model





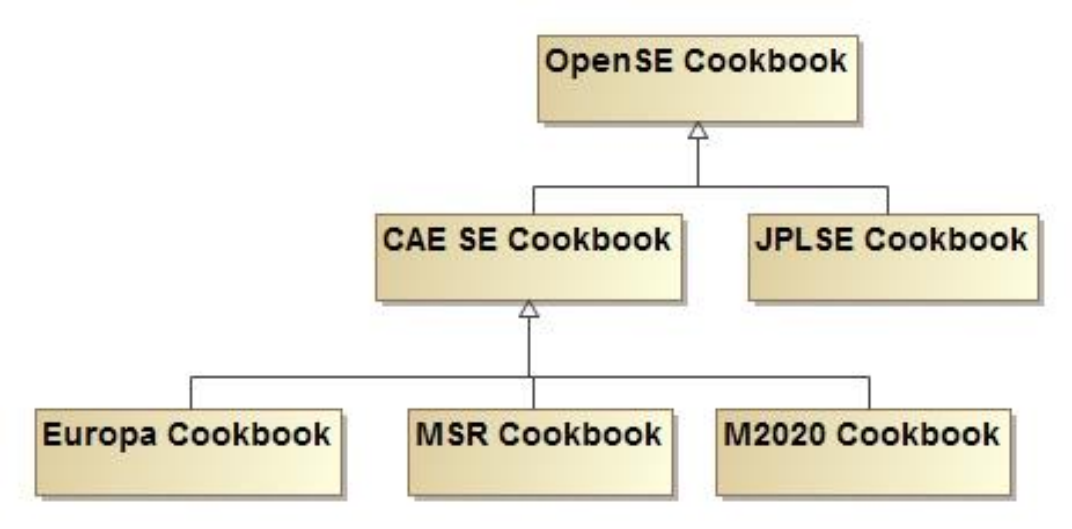


"Cookbook" for modeling methodology & patterns

Template models to be used by projects as a starting point, with recommended organization, model libraries, etc.

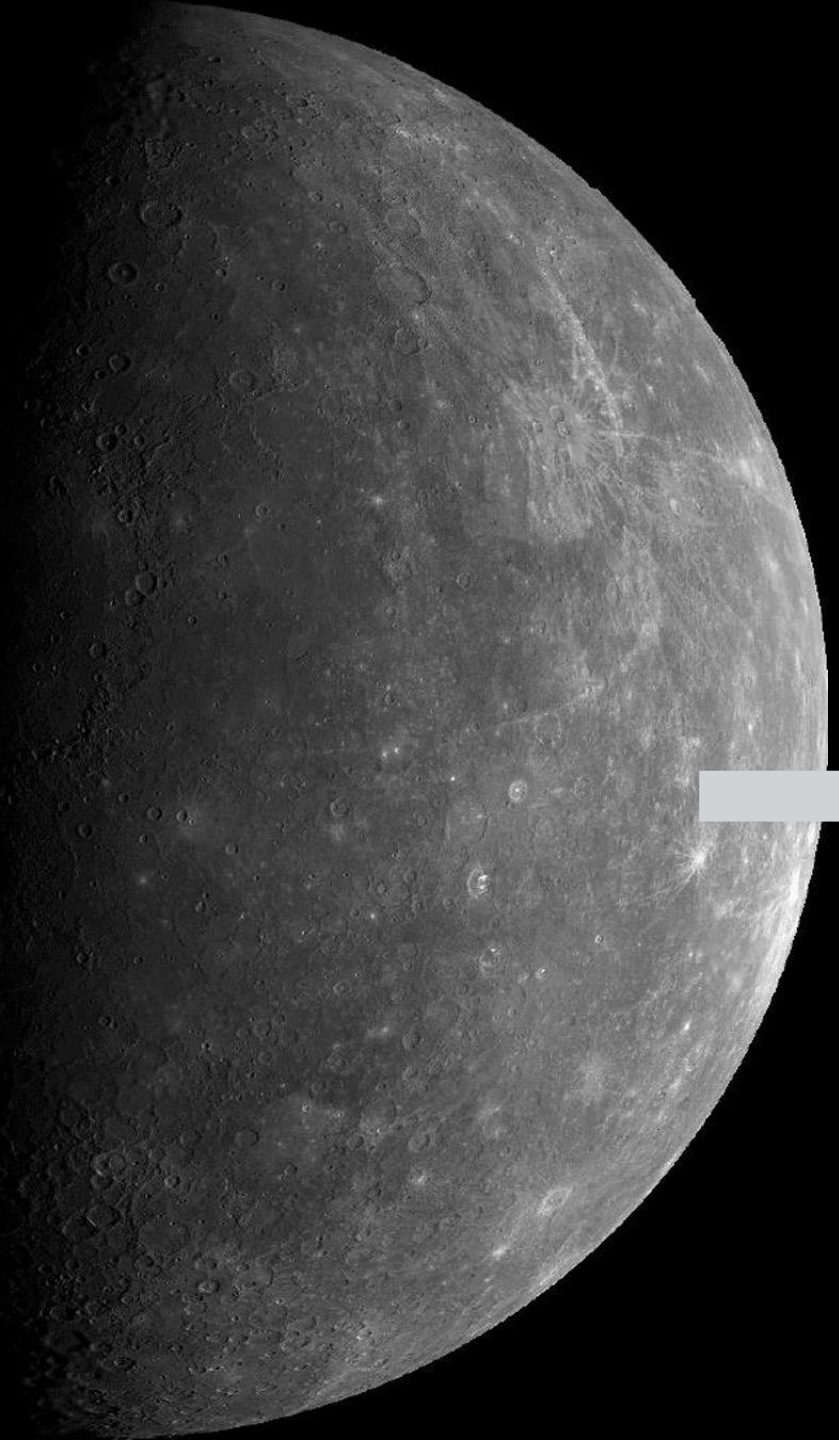
OpenSE Cookbook promotes re-use

- OpenSE Cookbook contributes to JPL institutional and project specific Cookbooks
- Project-independent modeling patterns as guidelines
- Project-specific modeling patterns for common modeling tasks



OpenSE Cookbook is used as reference

- OpenSE cookbook and TMT model used as reference model for the OMG SysML 2 standard
  - Demonstrate how SysML 2 will improve, simplify, change model wrt SysML 1.x
- Training material and knowledge transfer
- Promote standards and conventions
- Used by vendors as reference to test and evolve products



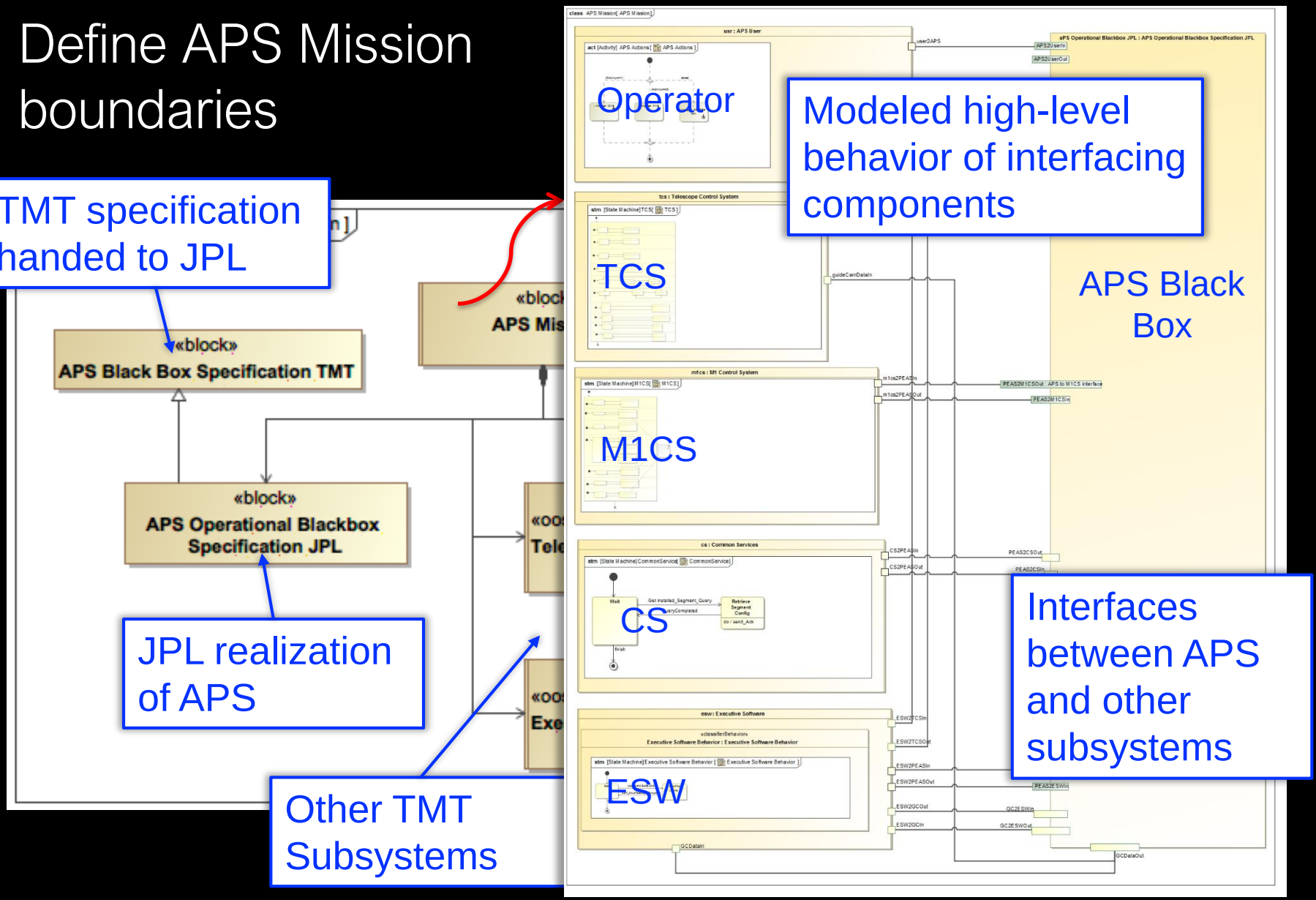
Agenda

- Introduction
- TMT and MBSE Approach
- Cookbook Principles
- Cookbook Examples

Systems Model is developed according to ESEM using Cookbook Patterns

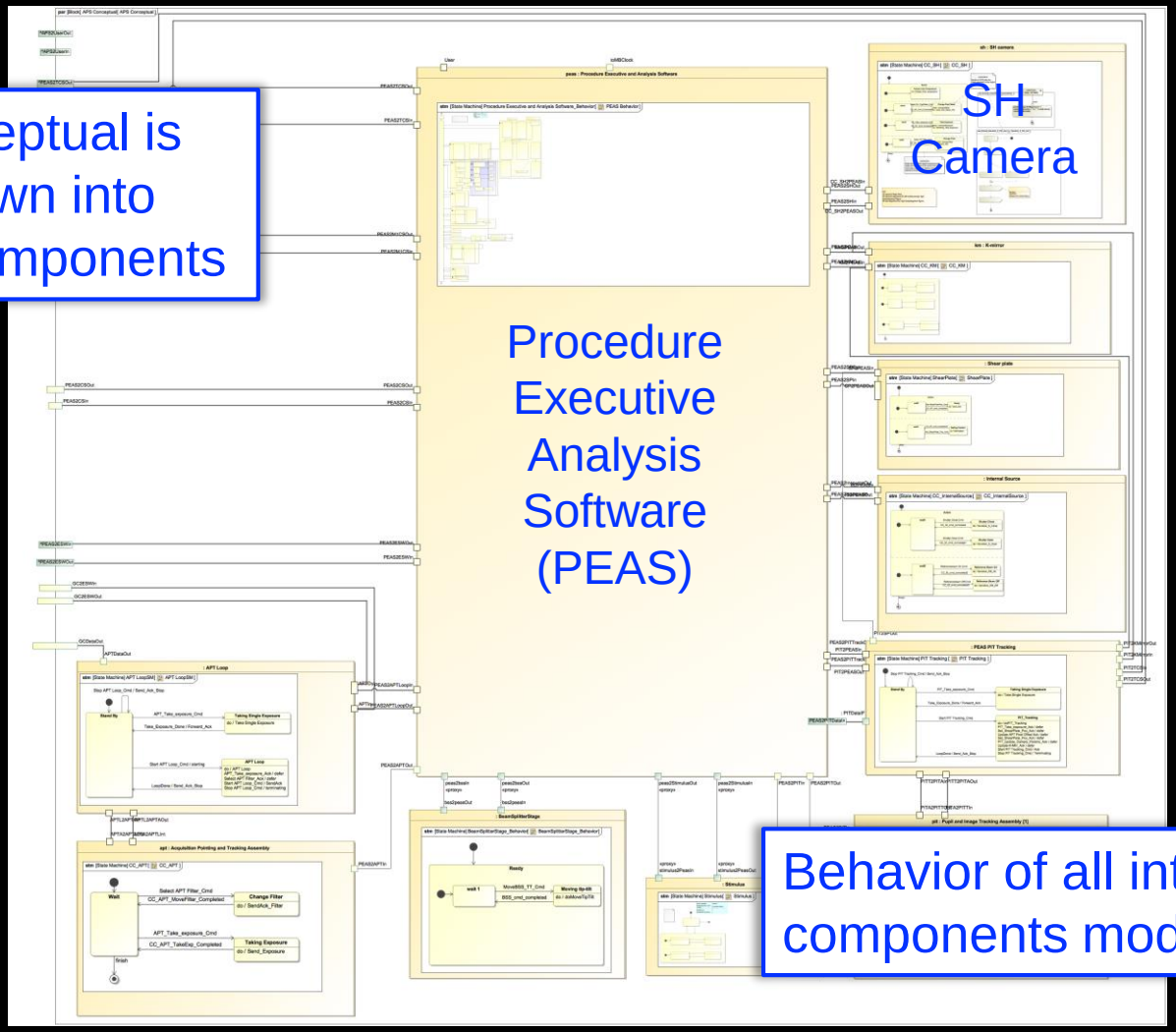
- Define APS Mission boundaries
- Elaborate Conceptual Architecture
- Capture Component Behavior and Characteristics
- Specify Interactions between Components
- Run Analyses

Define APS Mission boundaries



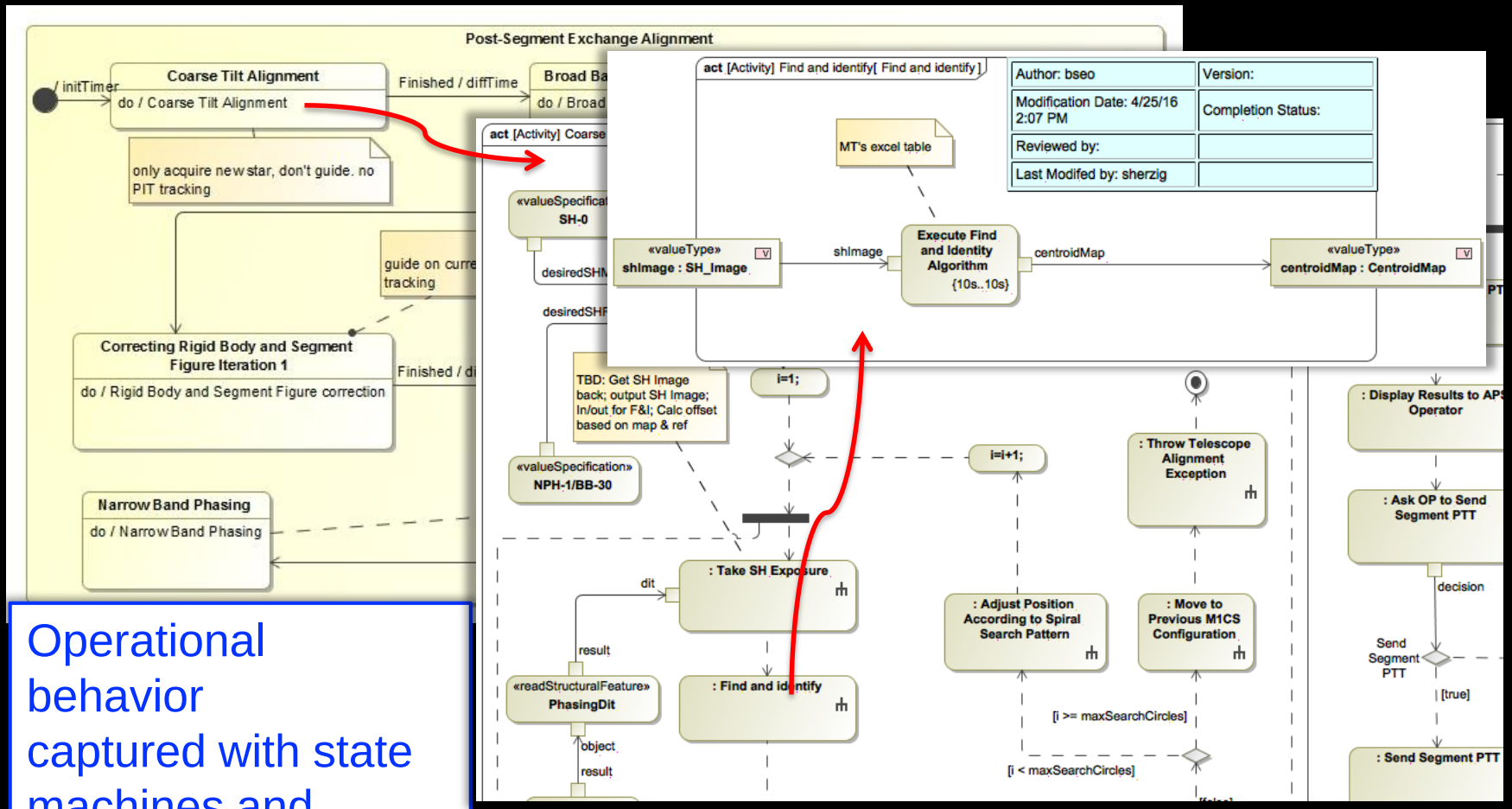
Elaborate Conceptual Architecture

APS conceptual is broken down into several components



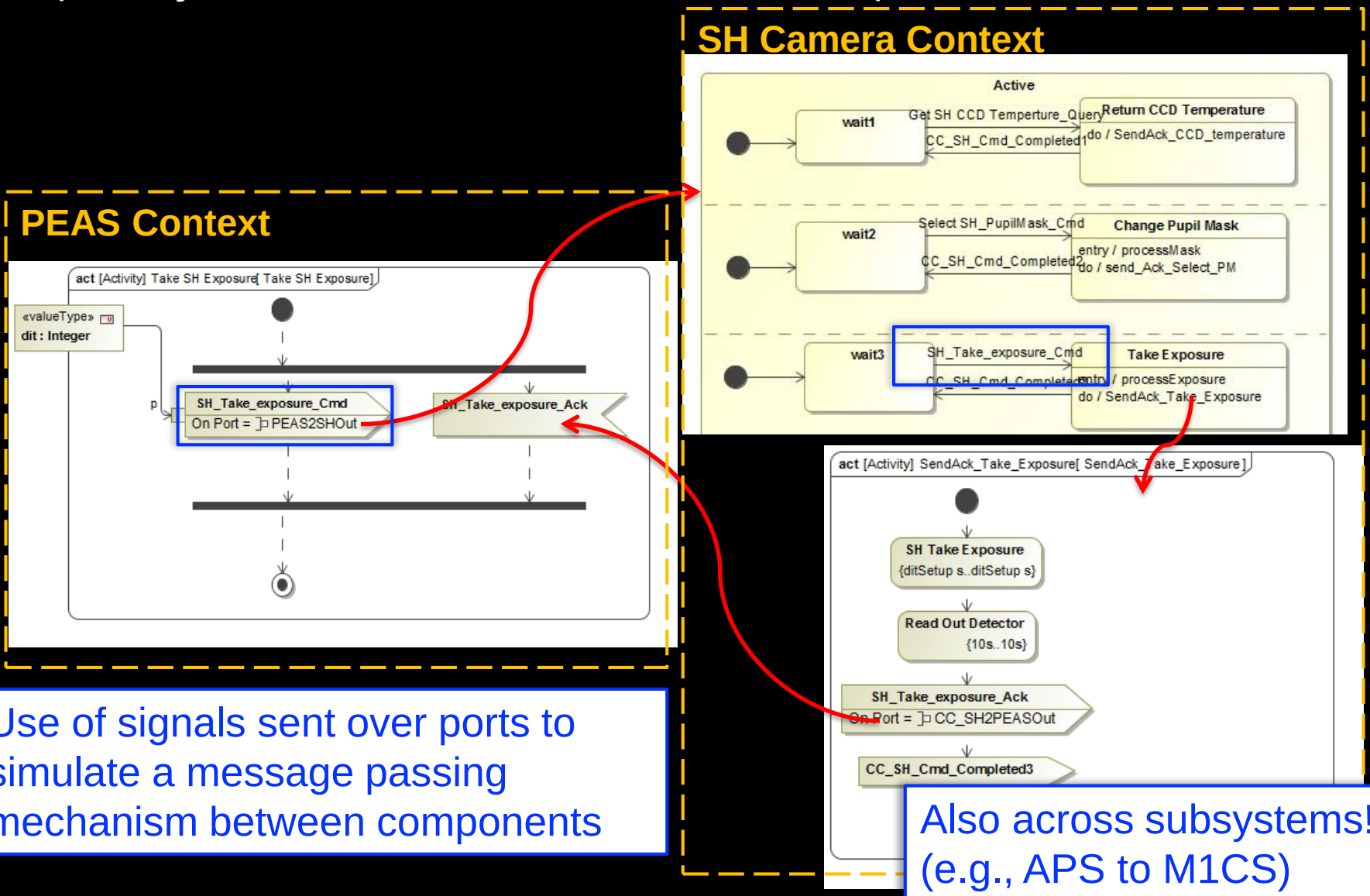
Behavior of all interacting components modeled

Capture Component Behavior and Characteristics



Operational behavior captured with state machines and activity models

Specify Interactions Between Components

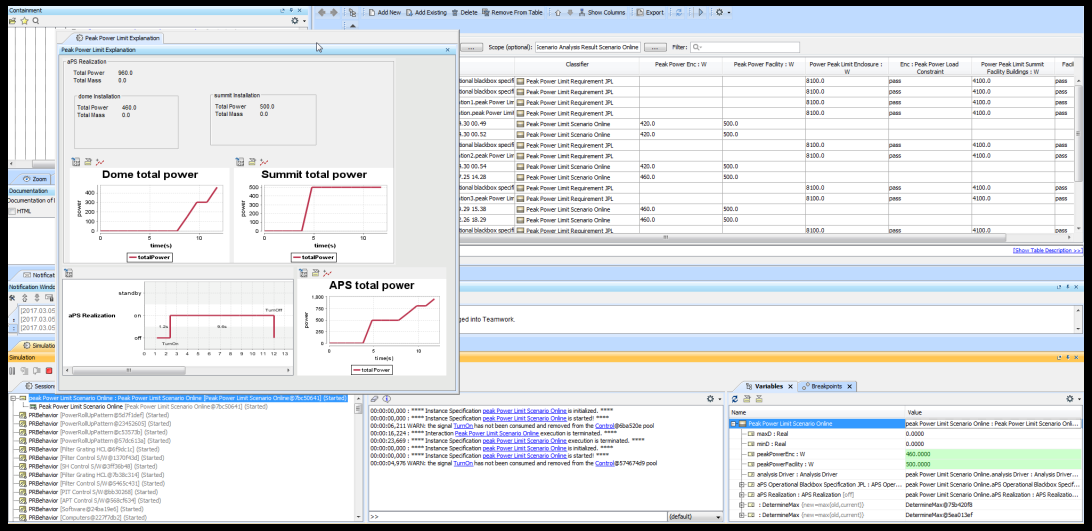


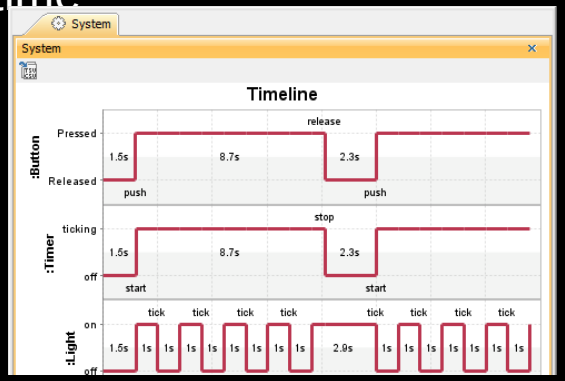
Use of signals sent over ports to simulate a message passing mechanism between components

Also across subsystems! (e.g., APS to M1CS)

Run Analyses

- Run a configured analysis with a simulation engine on the initial conditions to get the final conditions
- Produce the analyses declaratively, repeatably (in any system), without a single line of project-specific code -> reducing time and resources
- Produce the following views on final conditions
  - **Table** showing final analysis values (e.g. peak power) and the constraint's pass/fail status for each scenario
  - **Timelines**: state changes for components over time
  - **Value profiles**: total rolled up values over time



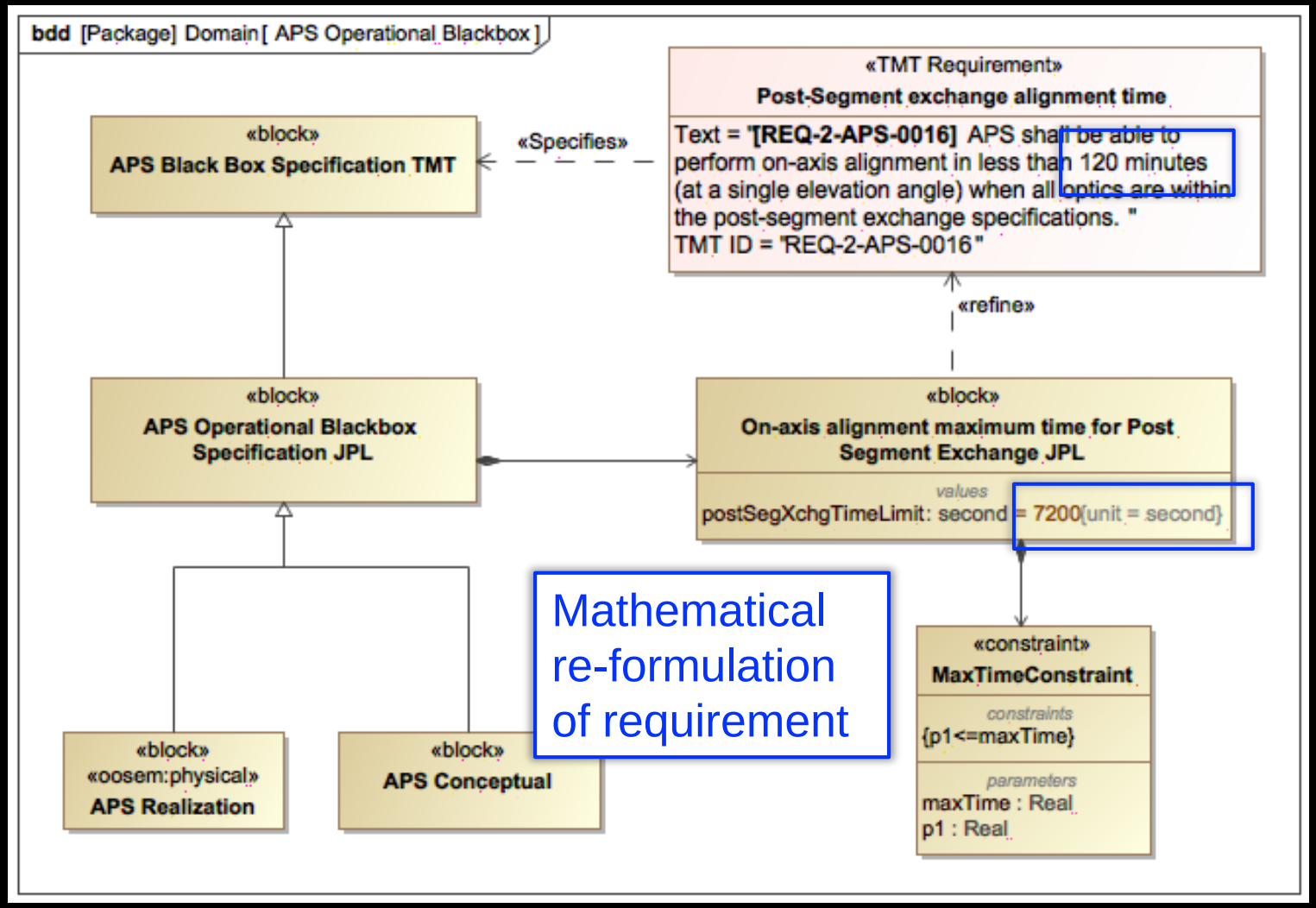


#	Name	Classifier	T Final : Real	Ph
1	calibrations Duration S	Calibrations Duration S		
2	calibrations Duration S	APS Conceptual		
3	calibrations Duration S	Procedure Executive ar	8466.0	11

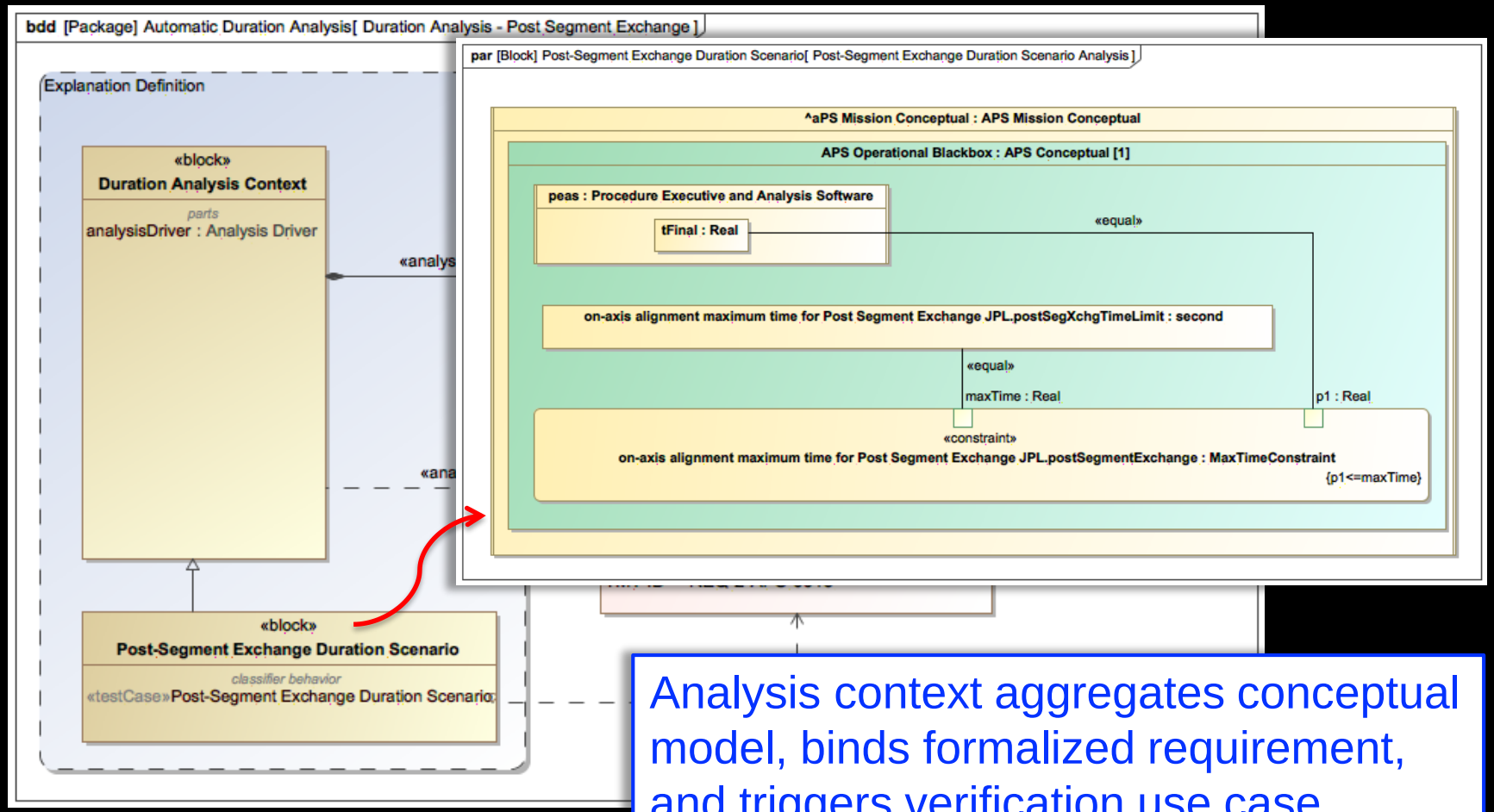
Requirements Verification

- Intent
  - Validate requirements, verify as designed system against requirements and publish analysis results
- Cookbook Volume
  - System Requirements Management
- Educational example
  - Autonomous Ferry Transportation
- Known Uses
  - APS - Post-segment exchange timing requirements
- Tooling
  - Cameo Systems Modeler and Simulation Toolkit, View Editor
- Notes
  - Property Based Requirements links Requirements Management and System Design

Formalizing Requirements with Property Based requirements

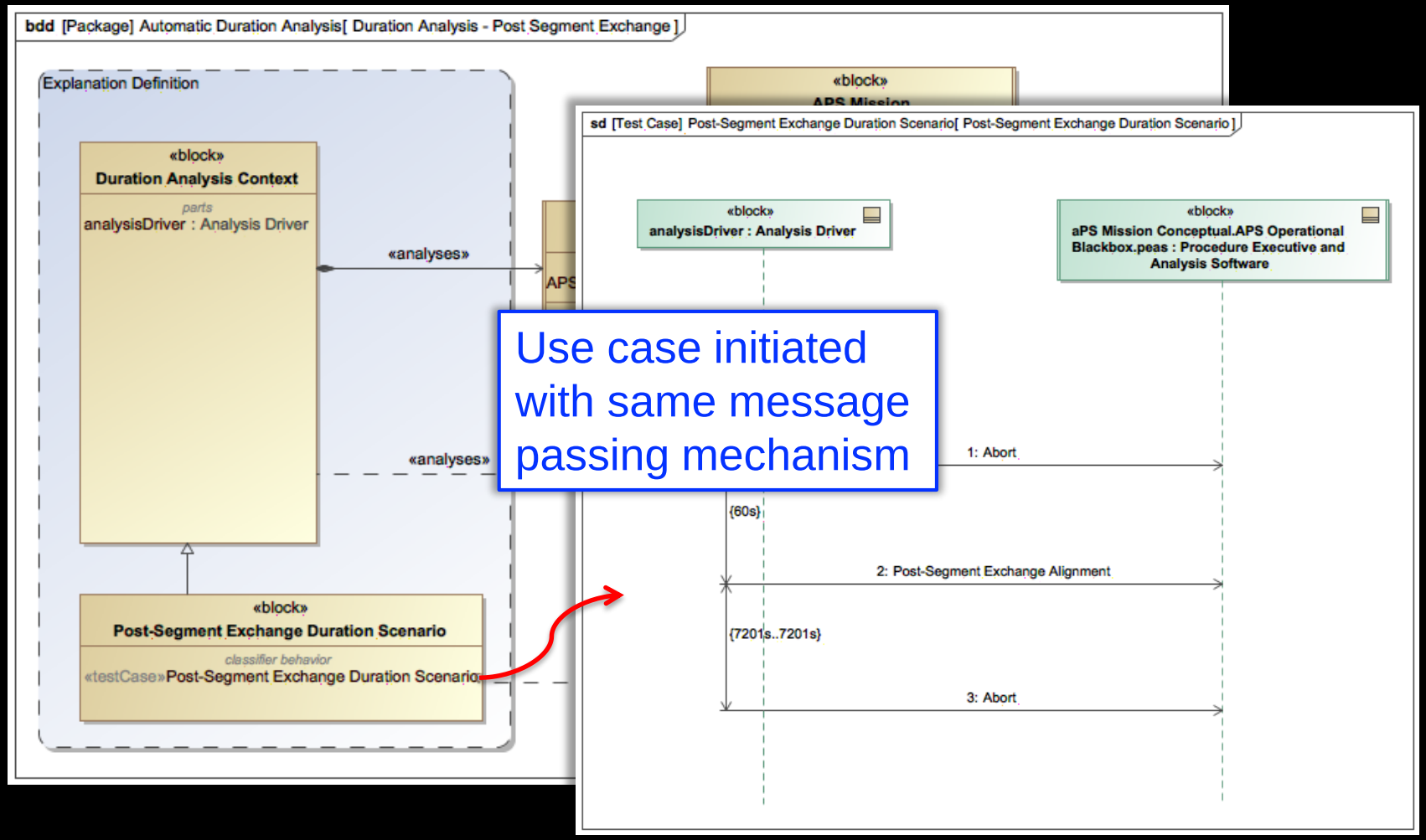


Verifying Timing Requirements by Simulation

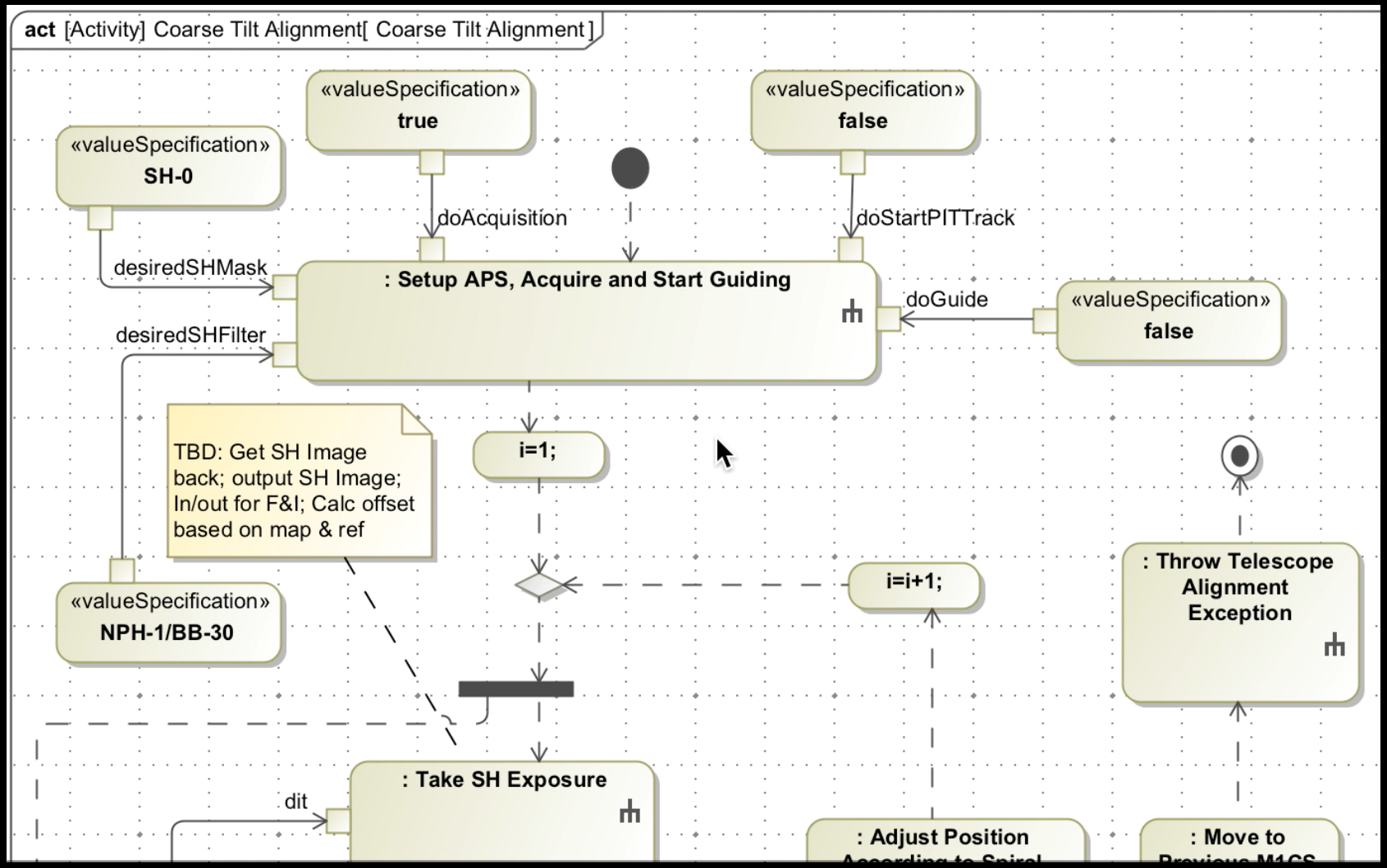


Analysis context aggregates conceptual model, binds formalized requirement, and triggers verification use case

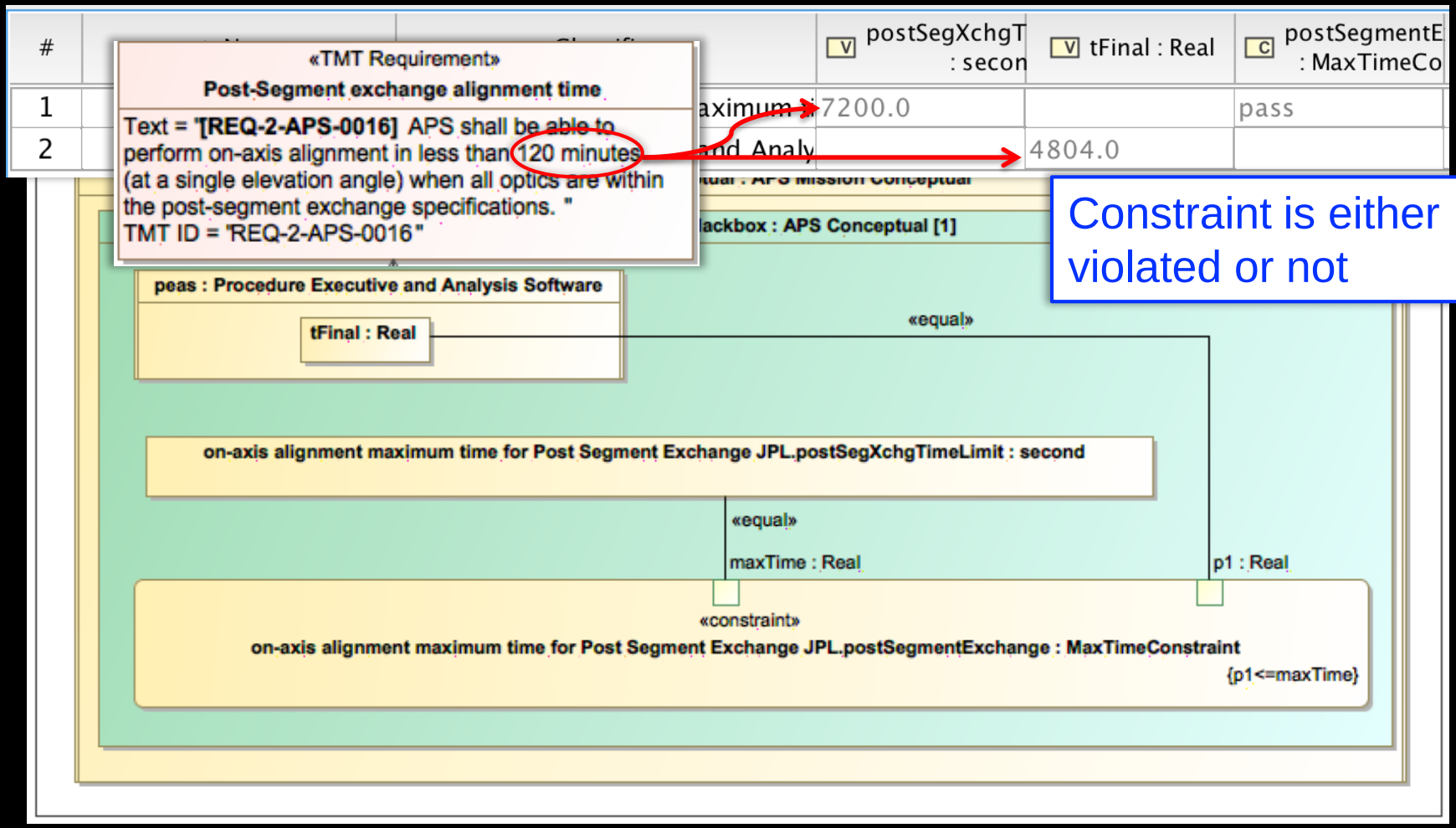
Verifying Timing Requirements by Simulation



Verifying Timing Requirements by Simulation



Verifying Timing Requirements by Simulation



Monte Carlo driven Analysis

- Intent
  - Estimate the characteristics and probability of a particular behavior using modeling anchored through data.
- Cookbook Volume
  - System Analysis Management
- Educational example
  - Quadrupedal Robot
- Known Uses
  - AO - Acquire a target with IRIS and NFIRAOS
- Tooling
  - Cameo Systems Modeler and Simulation Toolkit, View Editor
- Notes
  - SysML probability concepts and distributed properties capture operational knowledge in system model

Error Budget Management

- Intent
  - Manage error budgets of technical resources such as Mass, Power, Data
- Cookbook Volume
  - System Resource Management
- Educational example
  - Microscope
- Known Uses
  - APS - Alignment error of the M3 to APS interface
- Tooling
  - Cameo Systems Modeler and Simulation Toolkit, View Editor

Conclusions and Summary

- OpenSE Cookbook addresses SE concerns
- Built on proven patterns from TMT and APE
- Supported by tooling

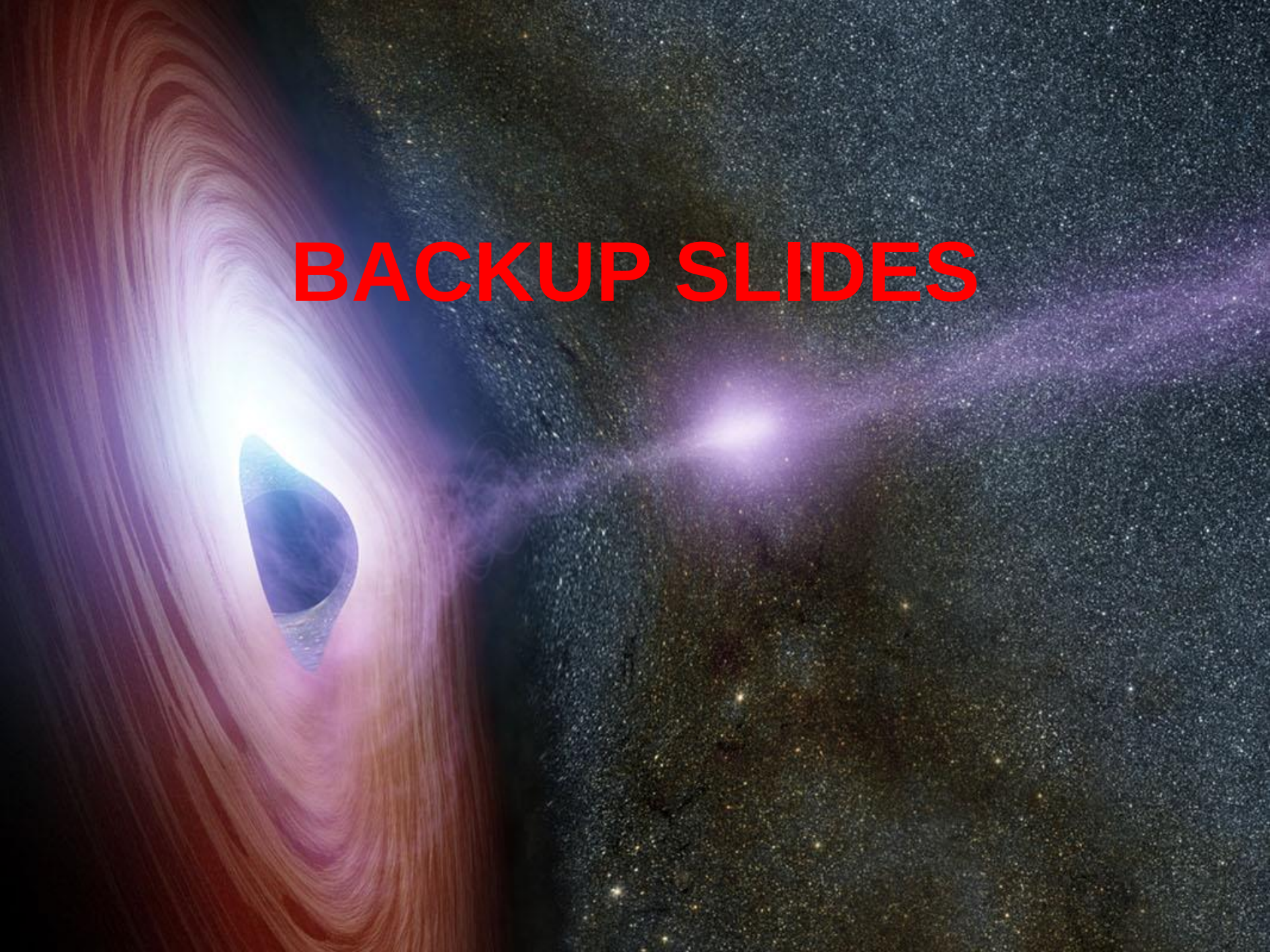
Acknowledgements

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

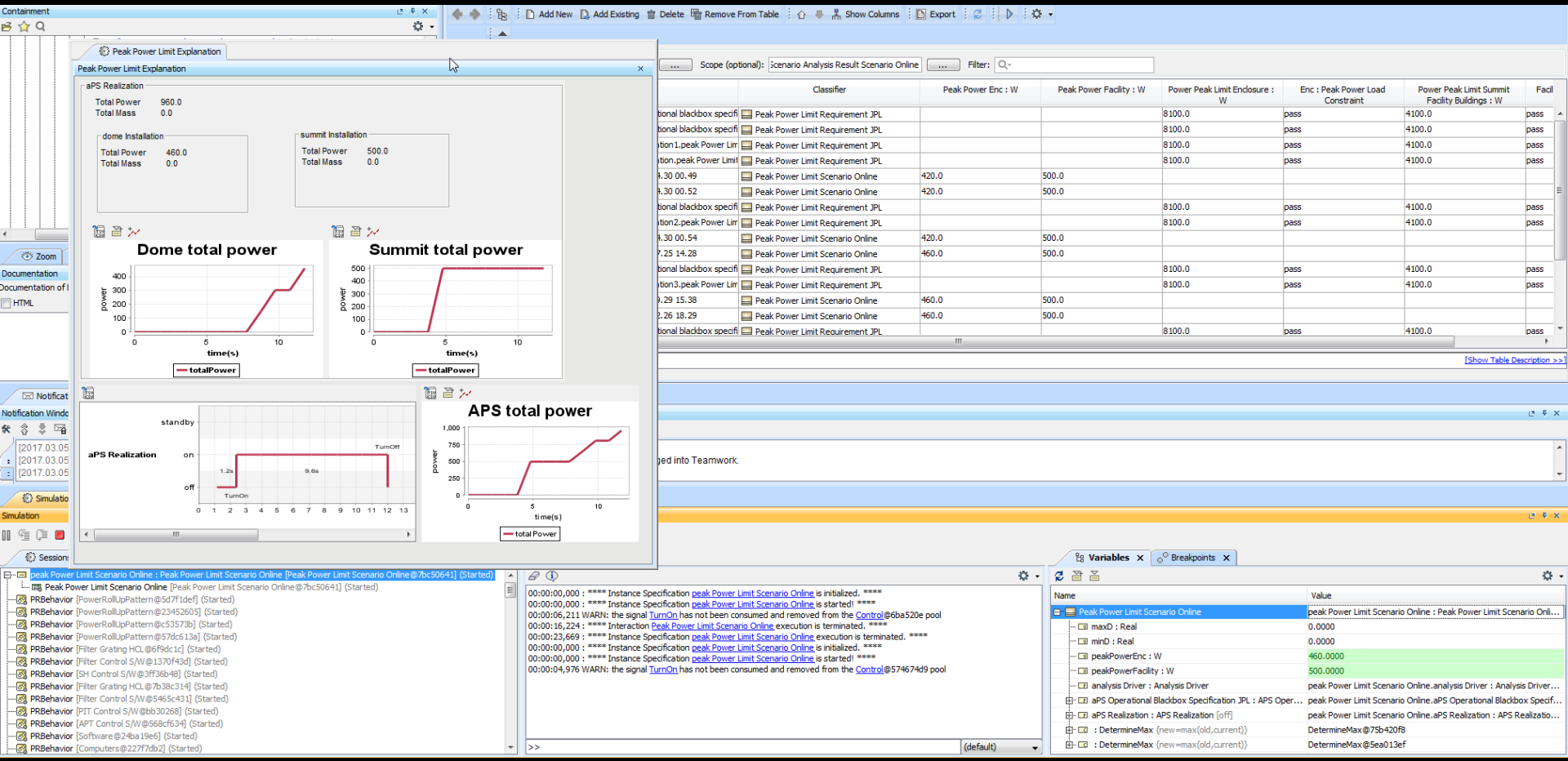
The TMT Project gratefully acknowledges the support of the TMT collaborating institutions. They are the Association of Canadian Universities for Research in Astronomy (ACURA), the California Institute of Technology, the University of California, the National Astronomical Observatory of Japan, the National Astronomical Observatories of China and their consortium partners, and the Department of Science and Technology of India and their supported institutes. This work was supported as well by the Gordon and Betty Moore Foundation, the Canada Foundation for Innovation, the Ontario Ministry of Research and Innovation, the National Research Council of Canada, the Natural Sciences and Engineering Research Council of Canada, the British Columbia Knowledge Development Fund, the Association of Universities for Research in Astronomy (AURA) and the U.S. National Science Foundation.

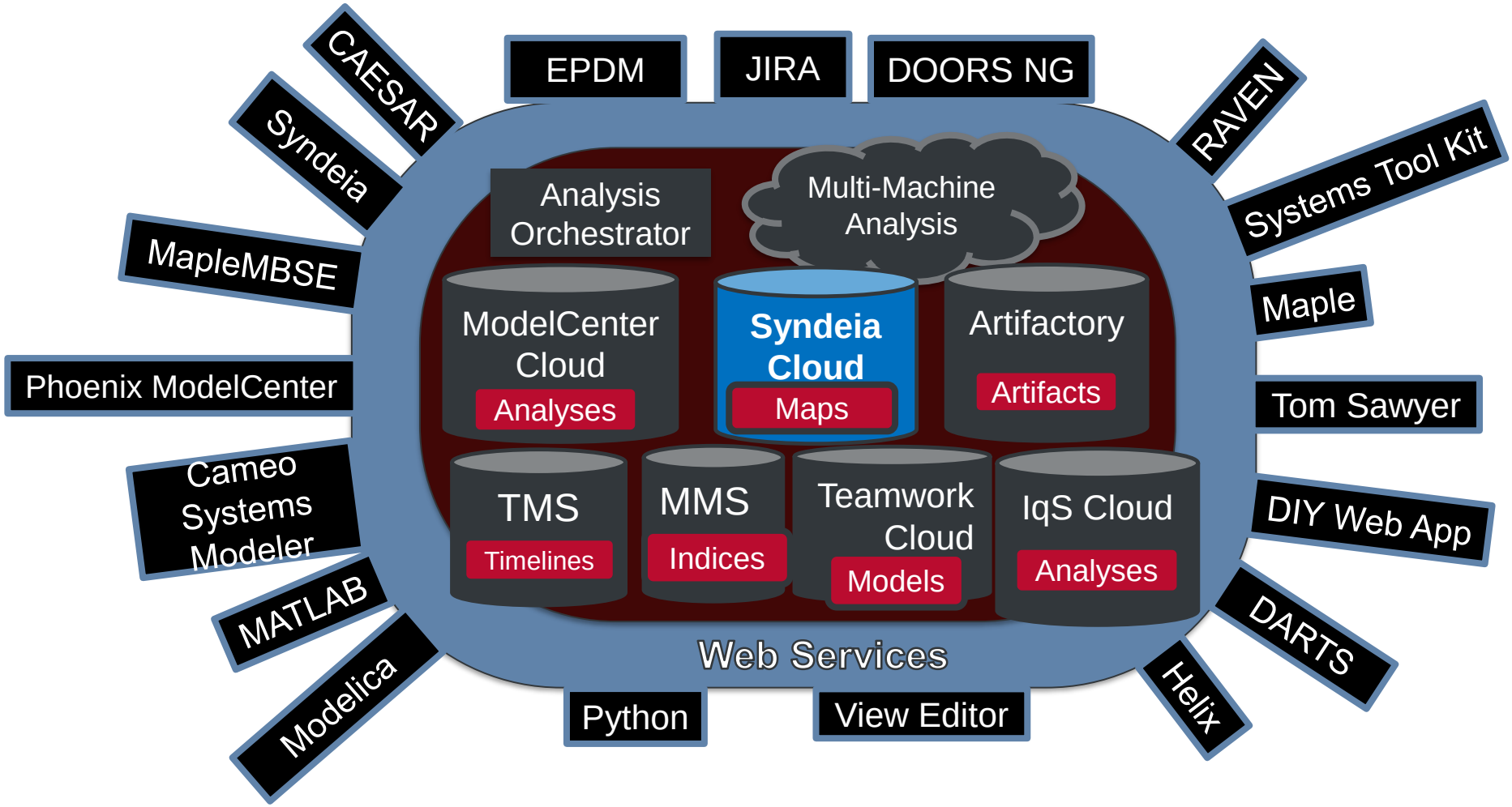
References

- INCOSE SE2 Cookbook: https://mbse.gfse.de
- OpenSE Cookbook: https://mms.openmbee.org
- Karban, R., Jankevičius, N., Elaasar, M. “ESEM: Automated Systems Analysis using Executable SysML Modeling Patterns”, (to appear in the proceedings of INCOSE International Symposium (IS), Edinburgh, Scotland, 2016.)
- Karban R., Dekens F., Herzig S., Elaasar M, Jankevičius N., “Creating systems engineering products with executable models in a model-based engineering environment”, SPIE, Edinburgh, Scotland, 2016
- Karban, R., “Using Executable SysML Models to Generate Systems Engineering Products”, NoMagic World Symposium, Allen, TX, 2016
- Open Source TMT model: https://github.com/Open-MBEE/TMT-SysML-Model
- Open Source Engineering Environment: https://www.openmbee.org
- Docgen, View&ViewPoints: https://github.com/Open-MBEE/mdk/tree/mdk-manual/src/main/dist/manual
- JPL Model-Based Systems Engineering Case Study: http://omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2017:iw_2017_open_mbee.pdf
- A Practical Guide to SysML, 3rd Edition, Chapter 17 by Friedenthal, Moore, and Steiner
- Zwemer, D., “Connecting SysML with PLM/ALM, CAD, Simulation, Requirements, and Project Management Tools”, May 2016
- https://www.jpl.nasa.gov/spaceimages/


BACKUP SLIDES

Power Analysis





OpenCAE DevOps